Presented at Sixth International Symposium of Robotics Research 1993
October 2-5, 1993; Hidden Valley, PA

Very Fast 3-D Sensing Hardware

Takeo Kanade

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3891

This paper describes two fast range sensing systems that are being developed at the Robotics Institute of Carnegie Mellon. The first system is a passive video-rate stereo machine, based on the multi-baseline stereo theory, whose expected performance will exceed that of existing stereo methods or medium-to-short distance scanning laser range finders in most categories: high throughput (1.96 Million points/second), frame rate (30 frames/sec), low latency (less than 2 msec after imaging), dense depth map (256x256), and high precision (7 bit) with uncertainty estimation. Another system is an active laser range-image sensor that acquires a complete 28 x 32 range frame in as little as one millisecond. Using VLSI, sensing and processing are combined into a unique sensing element that measures range in a fully-parallel fashion.

1 INTRODUCTION

Range information is crucial to many robotic applications. A range image is a 2-D array of pixels, each of which registers the distance to a point in the imaged scene. Many techniques for the direct measurement of range images have been developed[4]. Yet, range sensors developed so far are not yet fast, reliable, and inexpensive enough to be used broadly.

We have been developing two fast range sensing systems. The first system is a passive video-rate stereo machine, based on the multi-baseline stereo theory, whose expected performance will exceed that of existing stereo methods or medium-to-short distance scanning laser range finders in most categories: high throughput (1.96 Million points/second), frame rate (30 frames/sec), low latency (less than 2 msec after imaging), dense depth map (256x256). and high precision (7 bit) with uncertainty estimation. Another system is an active laser range-image sensor that acquires a complete 28 x 32 range frame in as little as one millisecond. Using VLSI, sensing and processing are combined into a unique sensing element that measures range in a fully-parallel fashion.

While one is passive and the other is active, both systems achieve the high performance by taking advantage of local parallelism which *can* be mapped into hardware. The following two sections will describe the current state of our development effort.

The first pan of this research was sponsored by the Avionics Laboratory, Wright Research and Development Center. Aeronautical Systems Division (AFSC), U. S. Air Force, Wright-Patierson AFS, OH 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597 and by the Department of the Army. Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709-2211 under Contract DAAH04-93-G-0428.

The second pan of this research was supported in part by the National Science Foundation, under grant MIP-8915969, and the Defense Advanced Research Projects Agency, ARPA Order No. 7511. monitored by the NSF under grant MIP-9047590.

2 VIDEO RATE STEREO MACHINE

2.1 Stereo: Unrealized Potential

Stereo ranging, which uses correspondences between sets of two or more images for depth measurement, has many advantages. It is passive and it does not emit any radio or light energy. With appropriate imaging geometry, optics, and high-resolution cameras, stereo can produce a dense, precise range image of distant as well as close scenes. Stereo performs sensor fusion inherently; range information is aligned with visual information in the common image coordinates. The same stereo algorithm *can* work with not only visible-domain CCD cameras but also other image sensors, such as infrared cameras, for night operation. Stereo depth mapping is potentially real-time and as fast as imaging; thus it does not have the problem of apparent shape distortion from which a scanning-based range sensor suffers due to motion during a scan.

Despite a great deal of research into stereo during the past two decades, no stereo systems developed to date have lived up to the potential as described above — especially in terms of speed, robustness, and accuracy. Existing "real-time" stereo

systems are not quite video rate. Often they provide only a small-foxmat (20x 20 to 64 x 64) depth *map* or sparse measurements (only at the edge points) at a low frame rate (several to 10 frames/sec) with low depth resolution (2 to 16 depth bins)[2, 5.9, 12,10, 15, 16].

2.2 Why Hasn't a Capable Stereo System Been Realized? — Illustration

The core difficulty of the stereo problem is that the correspondence problem is locally ambiguous. In other words, a point in the left image looks similar to many points in the right image when only local image information is considered. Humans seem to be successful in resolving this ambiguity so people assume that a clever algorithm will solve the correspondence problem unambiguously. Contrary to this common perception, a pair of images do not have enough information to uniquely determine correspondences, i.e., to give a unique depth map.

Figure 1 is a real-world example of such an ambiguous stereo pair: a shoe is placed with a carpet in the background. Since the carpet has a repetitive pattern, a grid point on the carpet in the left image has many equally good candidates of correspondence in the right image. One of the most common methods historically advocated to deal with this ambiguity is a coarse-to-fine strategy. Its basic idea is if one examines large chunk of the image (e.g., a region containing both shoe and carpet), a good match will be produced. So, gross matching is done first at a low resolution and then its result is used to limit the search range of matching at a higher resolution to reduce ambiguity. Almost all current near real-time stereo algorithms rely on this strategy. Figure 2 shows a depth map obtained by such a coarse-to-fine stereo matching method. The result looks reasonable, showing the shoe's shape on the carpet. However, the real situation is that the shoe is not "on" the carpet, but actually "above" it, as shown in the photo of Figure 3. The true depth map should be like the one shown in Figure 4, so Figure 2 is incorrect. What happened with the coarse-to-fine matching is that the unique matching at the shoe is propagated to the carpet to wrongly force its distance to be similar to that of the shoe. The coarse-to-fine strategy implicitly assumes surface continuity.

One can now see that exactly the Same (with no single-bit difference) image pairs can occur for the "shoe on carpet" and the "shoe above carpet" scene. In other words, the distance to the carpet cannot be obtained with the two images given in Figure 1! In most real-world situations, this difficulty may not show up as acutely as in this example, but quasi-regular patterns are common, such as building walls, trees, picket fences, and so on .

Historically, vision researchers have developed many tech-

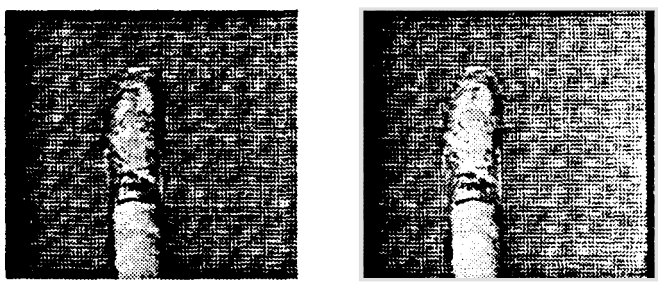

Figure 1: Right and left images of the "shoe-on-carpet" stereo pair.

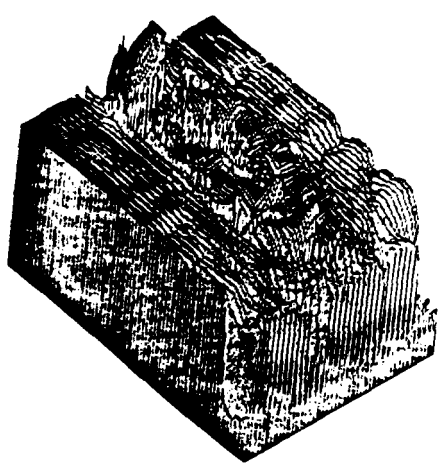

Figure 2: A depth map for the stereo pair of Figure 1 by a coarse-to-fine matching strategy.

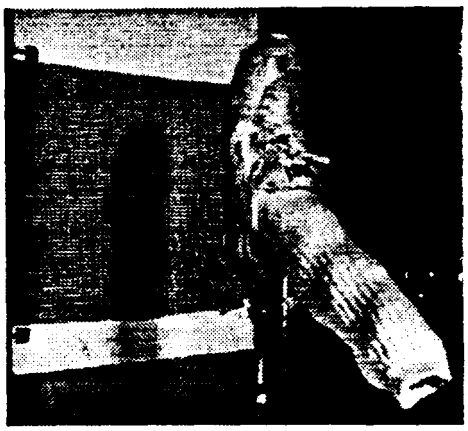

Figure 3: The true situation of the "shoe-on-carpet" scene: the shoe is actually above the carpet surface.

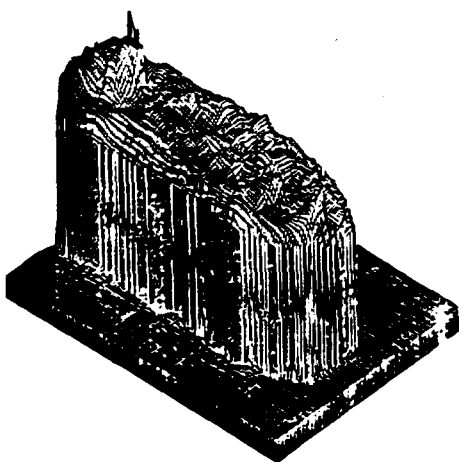

Figure 4: The true depth map — the height of the shoe is much higher than the carpet surface.

niques to alleviate the ambiguity problem, ranging from the coarse-to-fine strategy to the energy minimization and simulated annealing techniques. These are explicitly or implicitly based on surface coherency assumptions, such as continuity, planarity, membrane, and Markov Random Field. In terms of algorithms, they are global optimization seeking for the depth map which maximally satisfies certain assumptions as a whole. However, such global-optimization algorithms are not only computationally expensive, but also *can* lead *to* some artifacts in the results as illustrated above. Depth recovery cannot be a global optimization problem. Distance to a point in the space should be local; it should not depend on distances to other parts of the scene. What is required for a fast and reliable stereo machine is a non-optimization algorithm which operates solely on local information.

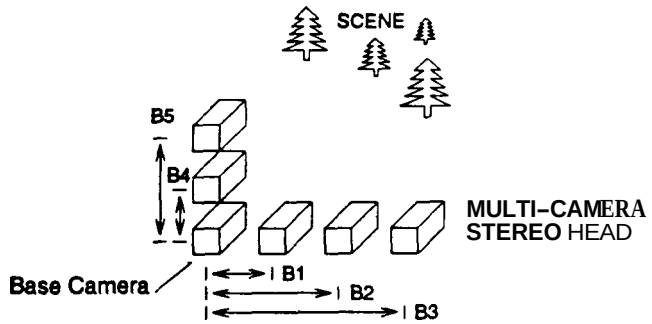

Figure *5:* Multiple baseline stereo setup.

2 3 The Multi-Baseline Method With SSD

Baseline and Matching

In stereo, the disparity measurement is the difference in the positions of two corresponding points in the left and right images. The disparity *d* is related to the distance *z* to the scene point by:

$$d = BF\frac{1}{z}. \qquad (1)$$

where B and F *are* baseline and focal length, respectively. This equation indicates a simple but important fact. The baseline length B acts as a magnification factor in measuring d in order *to* obtain z. The estimated distance, therefore, is more precise if we set the two cameras farther apart from each other, which means a longer baseline. A longer baseline, however, poses its own problem. Because a larger disparity range must be searched, there is a greater possibility of a false match. So a trade-off exists about selection of the baseline lengths between precision of measurement and correctness of matching.

The multi-baseline stereo technique being developed at CMU [13, 8] uses multiple images obtained by cameras which *are* laterally displaced (either or both horizontally and vertically) and provide different baselines relative to the base camera (see Figure *5).* Stereo matchings generated from several image pairs with different baselines are fused in such a way that information from pairs with a shorter baseline insures correctness of matching (i.e., robustness) and information from pairs with a longer baseline enhances localization (i.e., precision) of matching.

Sum of **SSDs**

The multi-baseline stereo method is based on a simple fact: if we divide both sides of (1) by B, we have:

$$\frac{d}{B} = F\frac{1}{z} = \zeta. \qquad (2)$$

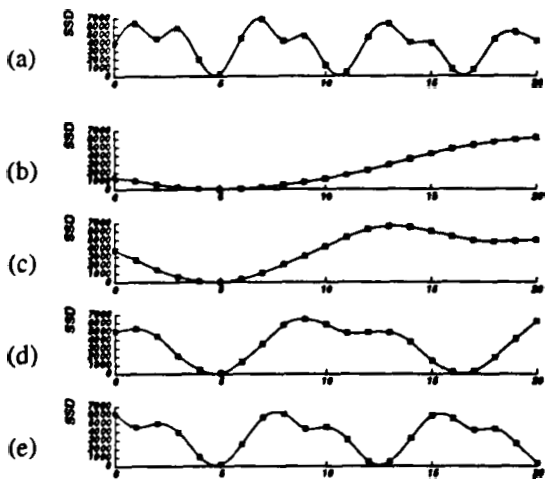

Figure 6: **SSD** values vs. inverse depth: (a) B = 8*b*; (b) *B* = b; (c) ***B* = 2*b*;** (d) ***B* =** 4b; (e) ***B* = 6*b*.** The horizontal axis is normalized such that 8bF = **1.**

This equation indicates that for a particular point in the image, the disparity divided by the baseline length (the inverse depth ζ) is constant since there is only one distance z for that point. Therefore if any evidence or measure of matching for the same point is represented with respect to ζ, it should consistently show a good indication only at the single correct value of ζ independently of B. Therefore, if we fuse such measures from stereo of multiple baselines into a single measure, we can expect that it will indicate a unique match position.

This fact can be best illustrated by the scene depicted in Figure 1. The grid pattern of the carpet is completely repetitive. ***So,*** the matching for a point in that region is inherently ambiguous. The **SSD** (sum of squared differences) over a small window is one of the simplest and most effective measures of image matching. For a particular point in the base image, a small image window is cropped around it, and **as** it is slid along the epipolar line of other images', the SSD values are computed for each disparity value. Such **SSD** values with respect to disparity for a single stereo image pair is shown **as** the top plot (a) of Figure ***6*. As** expected, it has multiple minimums and matching is ambiguous.

Imagine that we take multiple images of the scene with cameras displaced horizontally. We compute the **SSD** values from each individual stereo pair, and represent them **as** a function of the inverse distance ζ, rather than **as** that of the disparity d. The bottom four plots shown in Figure ***6*** are these functions, and we observe that all of them have

We use the Laplacian of Gaussian *(LOG)* filtered images instead of the intensity images to avoid the effect of intensity differences.

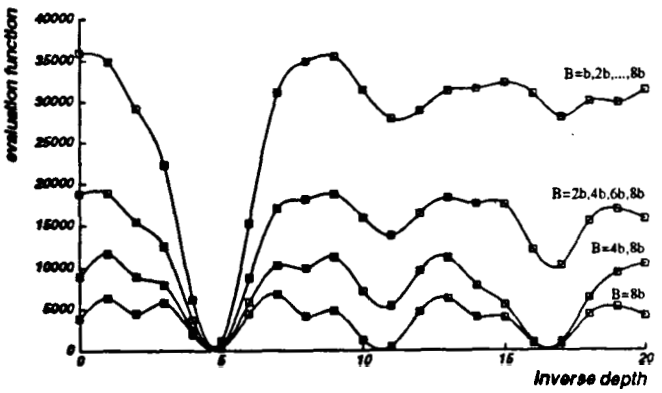

Figure **7:** Combining multiple baseline stereo pairs.

a minimum **near** ζ = ***5*** (the correct answer). We add up these SSD functions from all stereo pairs to produce the sum of **SSDs,** which we call **SSSD-in-inversedistance.** Figure **7** shows curves of the SSSD-in-inverse-depth for several stereo pairs. **The** bottom curve is obtained by a single baseline (i.e., **SSD,** the ***same*** **as those** in Figure ***6)*** and it shows multiple minimums. **As** the number of baselines increases to two, four and eight, the **SSSD-in-inverse-distance** has more clear and unambiguous minimum. Also, one should notice that the valley of the **SSSD** curve becomes sharper **as** more images are used. This means that we can localize the minimum position more precisely, thereby producing greater precision in depth measurement. The correct depth map shown in Figure **4** for the "shoe on carpet" Scene has been produced by this method using four images.

Kanade and Okutomi [8, 13] have proven that the SSSD-in-inversedistance function always exhibits a unique and clear minimum at the correct matching position. More specifically, suppose that the underlying image is periodic and ambiguous, like the carpet in Figure **1,** with period T, and that stereo pairs with baselines $B_1, B_2, \ldots, B_n$ are used. Then the SSSD-in-inversedistance is still periodic, but its period becomes:

$$LCM\left(\frac{T}{B_1F}, \frac{T}{B_2F}, \ldots, \frac{T}{B_nF}\right) \qquad (3)$$

where LCM() **means** the least common multiplier. **By** choosing appropriate B_i's we *can* make **this period** longer than the largest expected disparity, guaranteeing to have a unique matching, thus removing incorrect matching. Also, they have proven that the uncertainty of the measurement expressed by the variance decreases inversely proportionally to the sum of the square of the baseline lengths:

$$\sigma_n^2 = \frac{1}{B_1^2 + B_2^2 + \ldots + B_n^2}. \qquad (4)$$

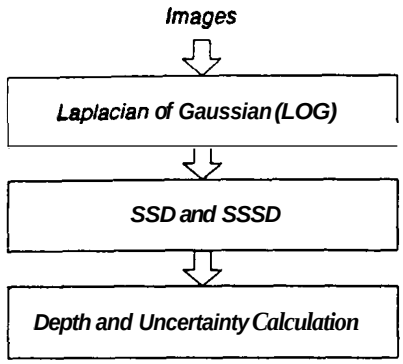

Figure 8: Outline of the multi-baseline stereo.

Obviously, this idea works for any combination of baseline directions. The computation is completely local, and does not involve any search, optimization, or smoothing. All the algorithm has to do is to compute the SSSD function and locate the single minimum for each pixel, which is guaranteed to exist.

Summary of the Algorithm

In summary our stereo method consists of three steps as shown in Figure 8. The first step is the Laplacian of Gaussian (LOG) filtering of input images. We use a relatively large (20x20) kernel for the LOG filter. This enhances the image features as well as removing the effect of intensity variation among images due to difference of camera gains, ambient light, etc. The second step is the computation of SSD values for all stereo image pairs and the summation of the SSD values to produce the SSSD function. Image interpolation for sub-pixel resampling is required in this process. The third and last step is the identification and localization of the minimum of the SSSD function to determine the inverse depth. Uncertainty is evaluated by analyzing the curvature of the SSSD function at the minimum. All these measurements are done in one-tenth subpixel precision.

Experiments

The algorithm has been implemented in C and tested with images from both indoor and outdoor scenes under a wide variety of conditions [11]. Generated dense depth maps are comparable to those by a state-of-the-art scanning laser rangefinder: a 256 x 256 range map for distance at 15 to 60 meters with one-tenth subpixel matching accuracy, producing approximately 7-bit (less than 1% error) equivalent output.

Indoor Calibrated Scene: This method has been tested with indoor scenes in the CMU Calibrated Imaging Laboratory for which we have a ground truth. The distance to the scene is typically 0.5 to 1m, the number of camera positions used ranges from 3 to 6, and the longest baseline length is only 15 to 38mm. The disparity range is 10 to 20 pixels. The distance error is typically less than 0.8%.

Outdoor Intermediate Distance Scene: Figure 9(a) shows one of the seven input images of an outdoor scene of bushes and parking meters. Figure 9(b) is an isometric plot of the disparity map produced by our algorithm. The parking meters, shrubs, no-parking sign (hard to see in the intensity image), automobile and building are clearly distinguishable. The distance to the automobile is approximately 15 m, and the far end of the building is at 34m.

Outdoor Field Scene: Our algorithm has been also tested with a large scale outdoor scene shown in Figure 10(a) at the Westinghouse Research Center in Pittsburgh where the CMU Planetary Rover, Ambler, was tested. The scene contains a grassy field with a line of trees at a distance of 60 m. Six images with horizontal displacements and six additional images with vertical displacement were used. The widest horizontal and vertical baseline in this set was 9cm. Figure 10(b) shows the disparity image. The noisy region is due to lack of features in the area of sky in the original image. This noise region is successfully detected as such by the uncertainty estimate. The plots in Figure 10(c) are the 3-D terrain profiles shown as height vs. horizontal distance along the vertical columns drawn in the figure above. We observe that the terrain features of the scene are correctly recovered; a flat and somewhat descending region at the front, a slope in the middle, and then a more gentle slope at the rear before the tree line. The measured distances to a few points in the scene have been verified to be correct to within 1%.

Stereo on Dante and NAVLAB 11: The multi-baseline stereo system has been put to practical use on a CMU robot named Dante, a large, 8-legged walking machine for the exploration of a live volcano in Antarctica. The system uses three cameras arranged along a 1-meter horizontal baseline. The system software was highly optimized in favor of speed in exchange for reduced precision. Output is a dense depth map of 256x256 pixels with 40 disparity levels in 7 seconds, which enables the robot to move slowly (2-3 MPH) through a field of obstacles. The stereo system has also been used to guide the CMU robotic truck, NAVLAB II.

2.4 Design and Development of a Video-Rate Stereo Machine

Based on our theory and experimental results with the multi-baseline stereo system, we have been designing a video-rate (30 images/sec), low-cost stereo vision system for robust and precise dense depth map generation. One of the features of this technique is that the algorithm is completely local in its

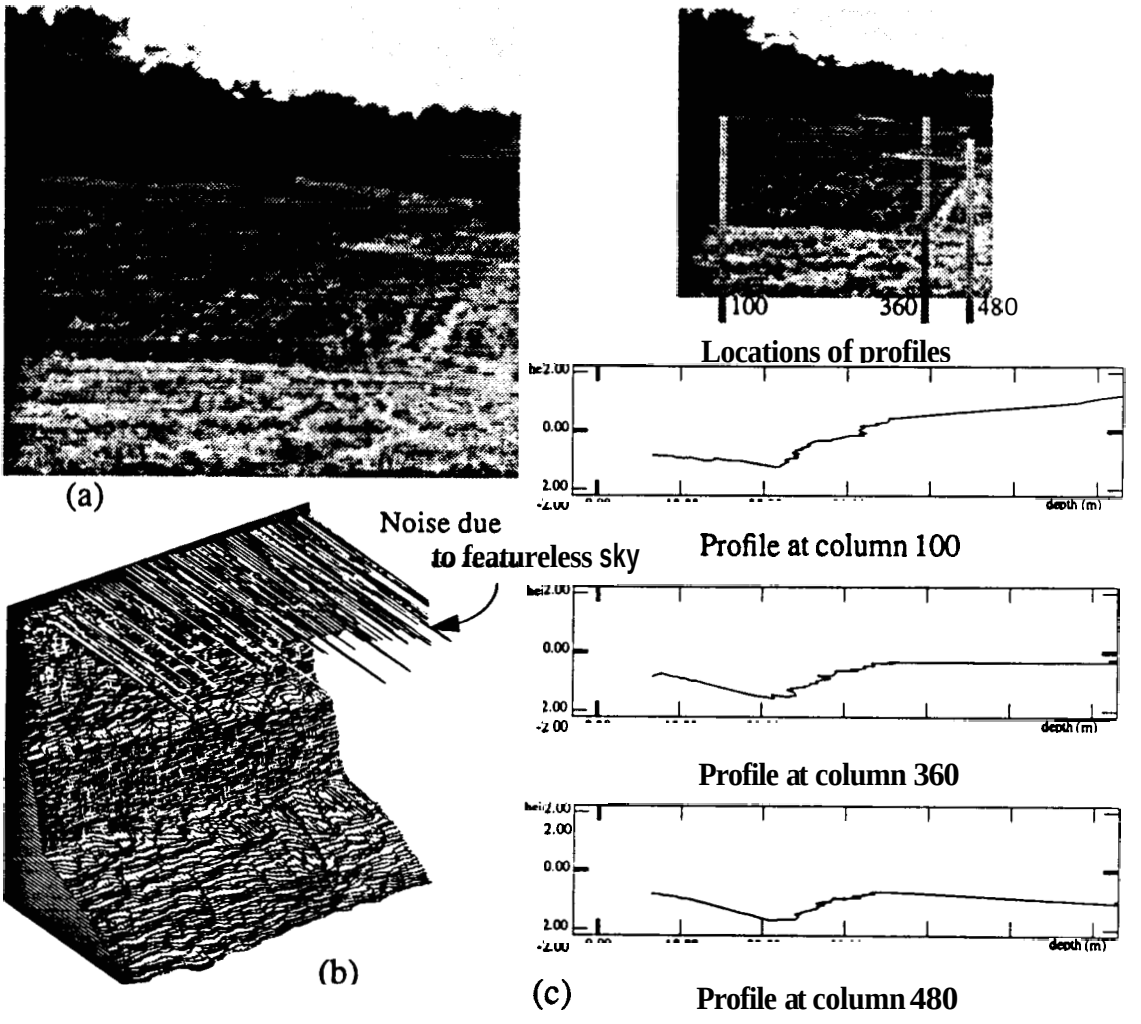

Figure 10: Grassy field scene: Input image (a), disparity *map* (b), and elevation profiles.

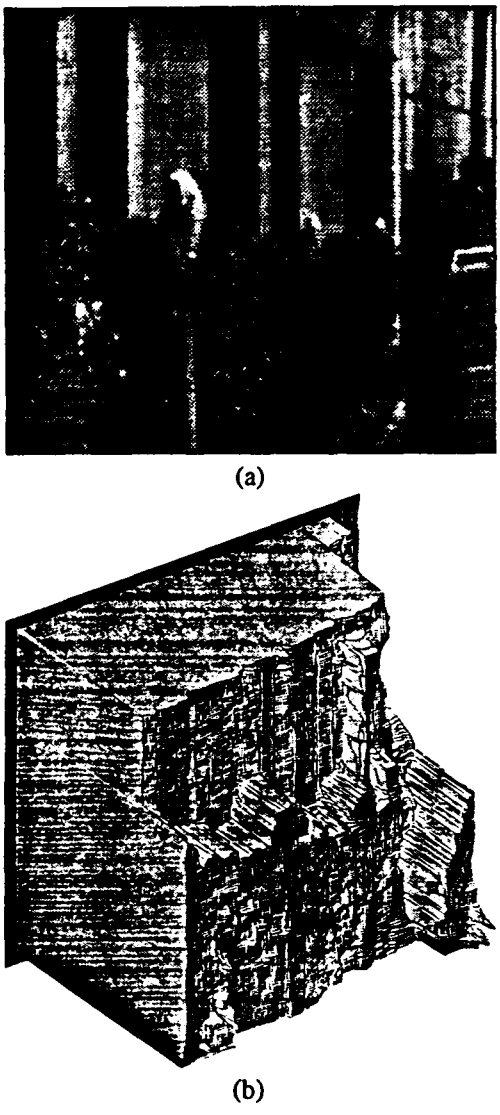

Figure 9: Parking meters scene: One of the stereo images (a) and disparity map (b).

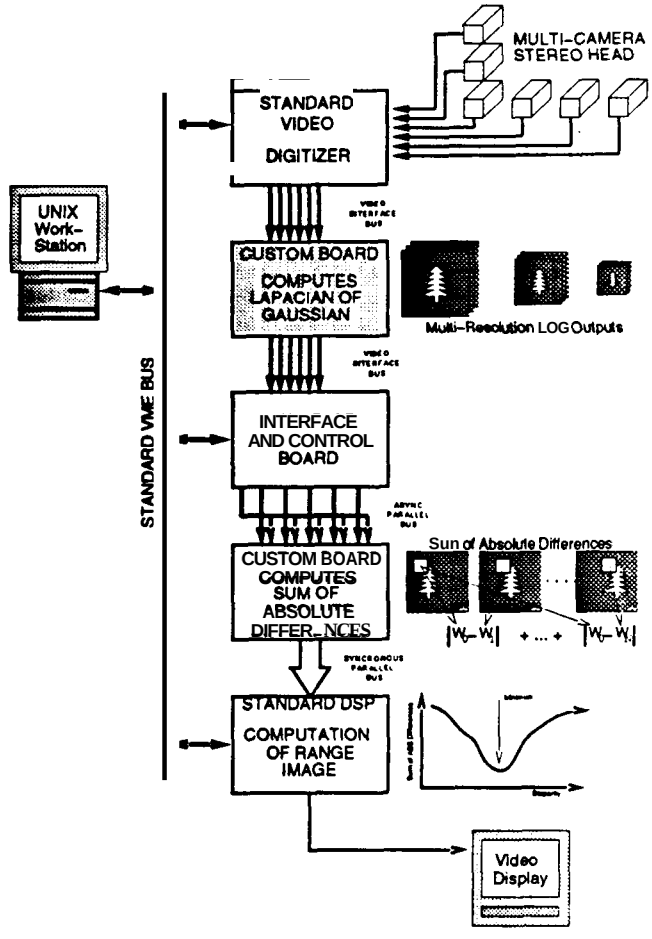

Figure 11: Configuration of CMU video-rate stereo machine.

computation. Computing the SSSD-in-inverse-distance function requires only a large number of local window operations appliedat each image position; no global optimization or comparison is involved. This is the most important for realizing a fast and low-cost stereo machine.

Overview

Figure 11 illustrates the configuration of the prototype system The system consists of four subsystems: 1) Multi-camera stereo head; 2) multi-image digitization; 3) Laplacian of Gaussian (LOG) filtering of input images in parallel; 4) parallel computation of SSD values and summation to produce the SSSD; and 5) subpixel localization of the minimum of the SSSD, and its uncertainty.

The video-rate stereo machine will perform these stages on a stream of image data in a pipeline fashion at video rate. The design performance of the system is as follows —

Number of cameras:	3 to 6
Output depth-image size:	256 x 256
Disparity levels:	20
Frame rate:	30 frames/sec
Throughput:	1.9M points/sec
Latency:	< 2 msec after digitization
Precision:	7 bits
Uncertainty estimation:	available for each pixel
Scene:	indoor and outdoors with natural lighting
Host:	Unix workstation with VME interface

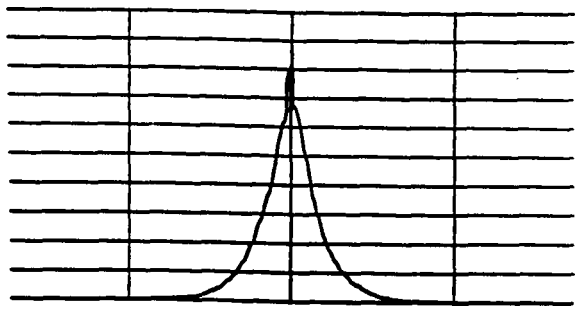

Figure 12: Histogram of a LOG-filtered image.

Theory for Machine Implementation

The basic theory requires some extensions to allow for parallel, low-cost, high-speed machine implementation. The two major ones are the use of small integers for image data representation and the use of absolute values instead of squares in the SSD computation.

Small Integer Representation: When an 8-bit input image is first filtered using a LOG filter, the dynamic range of the resultant image is expanded because of the relatively large (20 × 20) filter kernel and the wide dynamic range of the weights (the ratio of the largest to the smallest weights is 1 to 150). The current C language implementation of the LOG filter uses floating point representation of the result, but this does not be appropriate for a low-cost, real-time system We use a histogram equalization technique to map the LOG-filtered image to small-integer representation. Figure 12 shows a histogram of image data values of a LOG-filtered image of a "grassy-field scene" (Figure 10(a)). Since LOG is a band-pass filter, we see that the distribution concentrates near 0 and quickly decreases almost symmetrically as the absolute values become large. Large positive or negative values appear very rarely. The shape of the histogram is very similar for a wide range of images.

We assign more bits for values near zero and fewer bits for other values so that the mapped values are distributed uniformly. Our experimental results show that even when 4 bits are used, this method still produces disparity measurements which differ from the floating-point representation by less than 0.05 pixels in average. This is to be expected if we consider the following facts. When we use 4 bits to represent a LOG-filtered image, large numbers will have large errors since we assign fewer bits to them. However, since large values appear less frequently, they don't contribute much to the determination of the minimum of SSSD function. Moreover, useful matchings typically occur near edges along which LOG-filtered images have zero crossings and are therefore assigned a greater number of bits. Hence, this modification to the algorithm should still give results which are very similar to those produced by the floating point version.

Sum of Absolute Values: Another extension of the theory is to use the sum of absolute of difference (SAD) in place of the sum of squared difference (SSD). While this reduces the hardware parts count for the computation board, we also have verified that the performance does not differ. Use of the SAD computation together with small (4-bit) integer image representation will greatly reduce hardware requirements without sacrificing precision.

Design and Construction

Stereo-Camera Head: For the first prototype, we are building a 6-camera stereo head: four are arranged horizontally, and three vertically with one at the corder shared by both directions. The spacing between the cameras is 3 to 6 cm.

LOG-Filtering: We build six channels of a large kernel (20x20 equivalent) Laplacian of Gaussian filter by using GEC Plessey's convolvers. We do not employ techniques to decompose the $n \times n$ LOG filter into successive applications of $1 \times n$ and $n \times 1$ convolutions since this introduces a long latency which is an undesirable characteristic for a real-time control system.

SAD and SSAD Computation: This is the most computation intensive and the most critical part of the machine. If this computation were done in the most straightforward manner, for each stereo pair and for each point in the base image, we would need first to perform interpolation of image values ($P_{int} = 6$ operations) and then addition of absolute differences ($P_{sad} = 3$ operations). This happens for each point in the SSD window. The window is then shifted and this is again computed for each disparity value. Thus the total amount of computation per second would be:

$$N^2 \times W^2 \times D \times (C-1) \times (P_{int} + P_{sad}) \times F \qquad (5)$$

where N^2 is the image size, W^2 the window size, D the

disparity range, F the number of frames per second, and C the number of cameras. If we set $N = 256, W = 10, D = 20$, and $C = 6$, then this would be 172 GOPS. Actually, we can reduce the amount of computation to a manageable level by taking advantage of the fact that the computation includes a large amount of redundancy and that the image data is supplied as a stream. Variations of parameters, such as disparity range to search, the number of cameras, etc., are accommodated by rewriting the software for the micro sequencer.

Depth Extraction: An SSAD function for each point will be fed to TI's TMS320C40 DSP, which analyzes the function and locate the minimum position. The TMS320C40 DSP also computes the uncertainty of the result from the curvature at the minimum. Four DSP modules divide the task to achieve real-time performance.

Current **Status**

The design has been mostly finished, and the prototype video-rate stereo machine is being constructed. We expect the machine to be operational in early 1994. In addition to performing rigorous performance evaluations, we will field test the developed prototype real-time stereo machine in conjunction with the CMU Mobile robots, including Navlab, a lunar rover, and a helicopter.

3 A VERY FAST VLSI RANGE SENSOR

We have designed and fabricated a range sensor chip and has built, using this chip, a very fast range-imaging system [6, 7]. This range sensor acquires a complete frame (an array of 32 x 32 pixels for the moment) of 3-D data in a millisecond — two orders of magnitude faster that currently available range-image sensors. The sensor is based on the principle of cell-parallel light-stripe range imaging [1], a technique made possible by adopting VLSI computational sensor techniques.

3.1 Cell-Parallel Light-Stripe Range Imaging

Among many techniques for the direct acquisition of range images, the light-stripe methods have proven to be among the most robust and practical. We employ a novel "cell-parallel" algorithm that fundamentally improves the robustness of the stripe-based 3-D measurement process while increasing the 3-D measurement rate.

Figure 13 illustrates the principle of a light-stripe sensor. The scene to be imaged is lit by a stripe — a plane of light formed by fanning a collimated source. The stripe is projected in a known direction using a precisely controlled mirror.

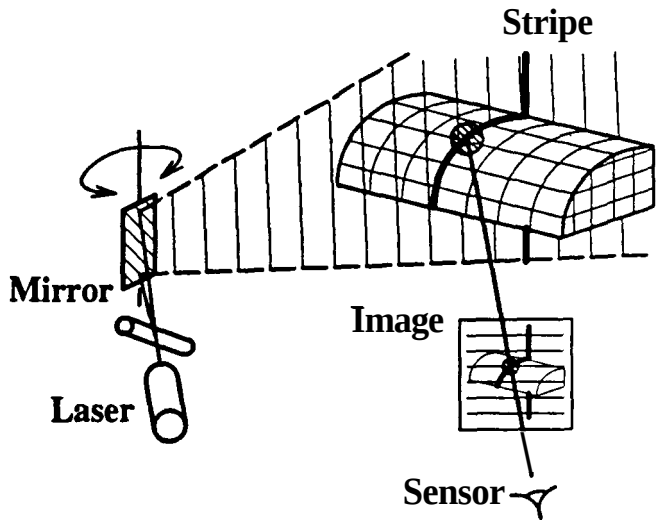

Figure 13: Light-stripe range sensor.

When viewed by an imaging sensor, it appears as a contour which follows the profile of objects. A traditional light-stripe range sensor takes a "picture" of the scene at a number of discrete stripe positions. The of the contour is extracted from each picture in the sequence in order to build a range image.

In contrast, the cell-parallel light-stripe range sensor that we have built uses a two-dimensional array of smart photosensitive cells, each of which "monitors" the change in intensity falling on the cell as the stripe continuously sweeps the scene. Every cell has circuitry that detects and remembers the at which it observes the *peak* incident light intensity during a sweep. Geometrically, a given cell predefines a unique line of sight (3-D line) and the "time-stamp" it records determines a particular orientation of the stripe (3-D plane). One 3-D data measurement is determined at the intersection of the stripe plane and the line of sight. Thus, sensing cells, working completely in parallel, acquire a complete range map from a single pass of the light stripe. The time required to build the range map is independent of the number of cells (i.e. — size of the range-image frame).

A practical implementation of the cell-parallel range imaging algorithm requires a — one in which optical sensing is local to the required processing.

3.2 Range Pixel Operation

Figure 14 summarizes the operation of cells in the sensor array. Functionally, each must convert light energy into an analog voltage, determine the time at which the voltage peaks, and remember the time at which the peak occurred. The straightforward "computation" extracts a single value from

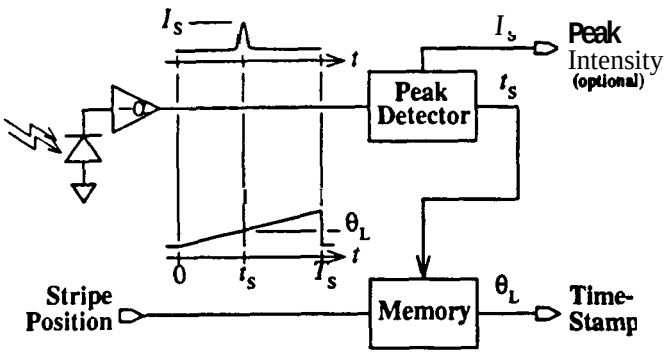

Figure 14: Range pixel processing.

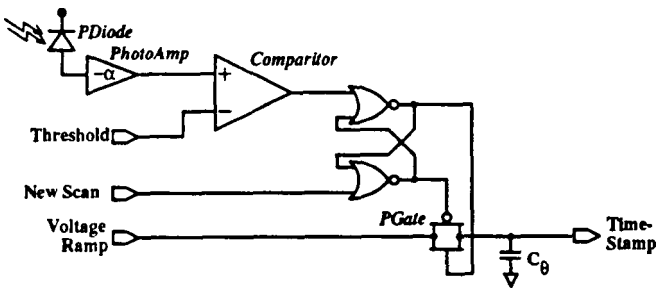

Figure 15: First-generation range cell — thresholding.

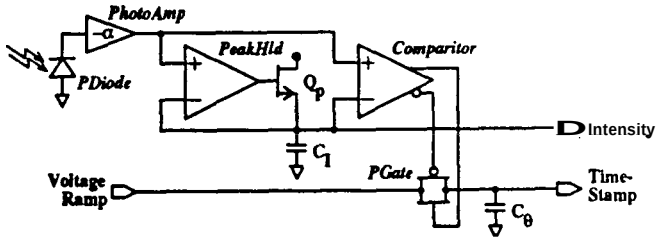

Figure 16: Second-generation range cell — "true-peak" stripe detection.

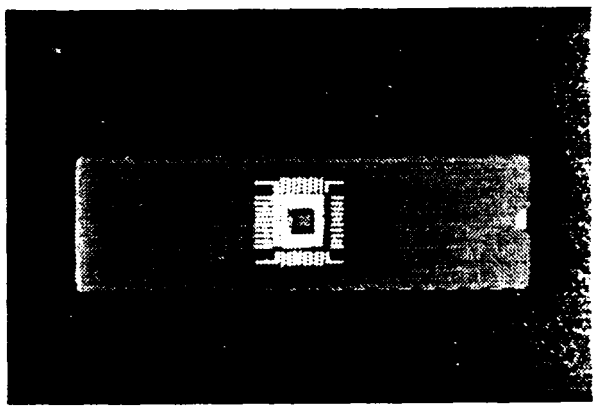

Figure 17: Second-generation range sensor IC chip.

the continuous output of a cell's photo-receptor — tremendously reducing raw transducer data bandwidth.

Range measurement is synchronized with stripe motion. When the stripe completes its scan, the sensor has recorded a range image in the form of held time-stamp values. A time-multiplexed read-out scheme is used to offload the analog data through a single chip pin. The time-stamp data, held as charge on capacitors within each cell, are gated onto a two-level on-chip bus using dual n/p-transistor pass gates. Dual-gate structures permit the use of rail-to-rail time-stamp voltages, maximizing the dynamic range of the analog time-stamp data. The charge is then integrated to produce a buffered voltage that is digitized by the chip interface circuitry.

3.3 Pixel Circuit Implementations

We have built, tested, and characterized two generations of VLSI range cell implementations. Circuit diagrams for these are shown in Figures 15 and 16. The first used thresholding and a single bit of digital storage to detect stripe passage. The second replaced thresholding with a me-peak detector. Both use an analog voltage ramp, broadcast to all cells, to measure time. A capacitor in each cell serves as an analog storage element for holding time-stamp values. The following paragraphs summarize the two designs, pointing out their strengths and weaknesses (as determined through laboratory experiments).

1st Generation — Thresholding Cell

The thresholding-cell design in Figure 15 compares the incident intensity level to a reference level set by off-chip circuitry. When intensity exceeds the reference, a flip-flop is triggered, latching the stripe position at that instant into a storage element within the cell.

Using the thresholding design, the first multi-pixel cell-parallel sensor chip was built. This chip consists of 896 sensing cells arranged in a 28 x 32 array. Each cell was 262 μm wide and 250 μm high, an area of 0.066 mm^2. The chip was fabricated using a 2 μm p-well CMOS, double-metal, double-poly process and measures 9.2 mm x 7.9 mm (width x height). Of the total 73 mm^2 chip area, the sensing cell array takes up 59 mm^2, read-out column-select circuitry 0.37 mm^2 and the output integrator 0.06 mm^2. The remaining 14 mm^2 is used for power bussing, signal wiring, and die pad sites.

The Circuitry needed to implement threshold-based peak detection is straightforward, an advantage of the approach. It is also bistable and so commits rapidly once a decision is

made. One disadvantage is that thresholdingdoes not find the true-peak position. However, our experiments have shown this is not a severe disadvantage in practice. The primary drawback is in setting the threshold. The threshold level is common to all cells and so an "appropriate"level must be found. Unfortunately, there is no single level which is suitable for all cells. Comparator offsets from input differential-pair mismatch causes a threshold which works for one cell to be too high for another.

2nd Generation — Peak-Following Cell

The second-generation circuit, shown in Figure 16, performs peak detection by continuously comparing the incident intensity with the highest intensity level seen since the start of the scan. Rising intensity input transitions are tracked on capacitor C_l through the source-following transistor Q_p. No path is provided for C_p to discharge when photo-receptor output transitions downward. At the end of a scan, the largest intensity reading observed will be held. As the current input intensity falls below this held value, the comparator transitions, recording the pixel's time-stamp value on capacitor on C_θ.

The sensor array built using peak-following has 1,024 pixels, arranged in a 32 x 32 grid. The cell dimensions are 216 μm x 216 μm, 40% smaller in area than the cells of the first-generation sensor. However, much of the reduction in cell area was due to an improved readout sequencing scheme, with photo-receptor area and active circuit area remaining fairly constant. Figure 17 is its photograph.

The peak-following design eliminated the need for a single global threshold — the major drawback of the first-generation cells. In addition, the peak-intensity value held within the second-generation cell is an important artifact of the new design. The peak-intensity data is provided as an additional chip output and is read from the chip in parallel with the raw range image. The intensity data adds a new dimension to the sensor — a direct measure of scene reflectance properties at the stripe wavelength. The intensity image is alignedperfectly with range readings from the array and is useful in range system geometrical calibration.

In device testing, the second-generation cell achieved improved sensitivity and tolerance of ambient lighting. It operates in the presence of bright indoor lighting that was a problem for the first-generation cell. On the negative side, experiments with the current version of the second generation sensor indicate that because the peak detector has relatively low gain, changes in the reflected stripe intensity result in a slight shift in the time at which the peak is detected, and hence in the measured range. We plan to redesign the peak detector to employ positive feedback which will overcome this problem. In addition, we have found that, once a multi-faceted scanning mirror is installed, the readout speed will be limited by settling time of the op-amp pad driver. We are in the process of redesigning the output pad drivers for faster operation.

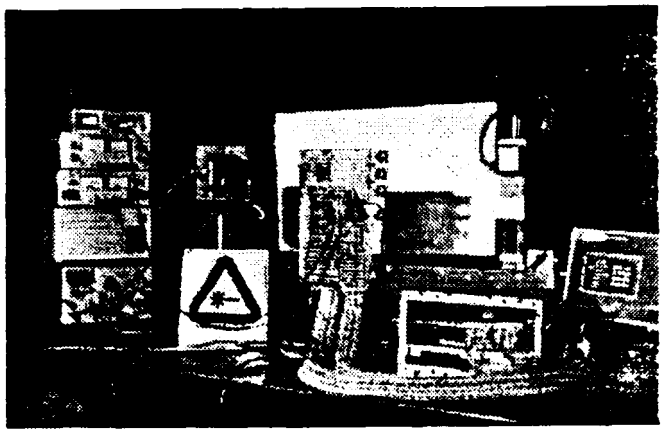

Figure 18: Prototype cell-parallel range image sensor.

3.4 Prototype Range-Imaging System and Experimental Results

In order to test and characterize our VLSI sensor, custom system hardware, software, and calibration algorithms were developed. Major hardware components built included a precise stripe generation and sweep assembly, sensor imaging optics, and interface electronics.

System Configuration

The prototype range-imaging system we have built is shown in Figure 18. The sensor chip is mounted on the circuit board visible in the upper right, behind the lens which forms an image on the sensor. Shown to the left of this, in the center of the picture, is the stripe-generation assembly. Light from a 30mW near-infrared (780 nm) laser-diode source is collimated and then fanned into the stripe using a cylindrical lens. In the prototype system, the stripe is swept through 40° using a spinning single-faceted mirror. The time-origin and duration of each sweep is measured using start-of-scan and end-of-scan detectors placed in the path of the beam. A 300 mm separates the stripe's axis of rotation and the optical center of the sensor assembly. Sensor calibration and testing is accomplished using targets moved using an accurate three degree-of-freedompositioner (not visible in the picture).

Performance

Spatial Resolution:	32 x 32
Acquisition Time:	< 1ms
Frame Rate:	> 250 frames/s
Operating Distance:	350 to 500 mm
Accuracy:	< 0.5 mm
Repeatability:	< 0.5 mm
Reflectance Image:	Available

Table 1 : Range sensor performance summary.

The second-generation VLSI range sensor acquires a complete range image in a millisecond. Accuracy and repeatability of the 3-D data has been measured experimentally to be 0.1% or better. Complete results of sensor characterization are summarized in Table 1. For comparison, Rioux and others at the National Research Council Canada (NRCC) have built one of the best range-image sensors to date[3], based on the synchronous scanning method. The NRCC sensor acquires frames of range data in 33 ms, accurate in depth to 0.4 percent. As is typical of systems that measure range data sequentially, frame time increases with spatial resolution. The frame time of our cell-parallel implementation, in contrast, is independent of the range image spatial resolution.

The quality of the range data produced by the cell-parallel range sensor was measured by holding a planar target at various known world-r positions with a three degree-of-freedom positioning device. These experimental results characterize the mean measured range value to be within 0.5 mm at the maximum 500 mm z — an accuracy of 0.1%. The aggregate distance discrepancy between world and measured range values remains less than 0.5 mm over the entire 360 mm to 500 mm z range.

The rapid 3-D-frame acquisition time allows us to capture a sequence of range images of a moving object without distortion. Figure 19 shows a "range movie" of one such object — a moving hand. These images were taken with the second generation 32 x 32 sensor chip on 15ms intervals using a 1ms stripe scan.

Recently we demonstrated a real-time pose estimation task using the VLSI range-image sensor which performs full 3-D pose estimation of a single arbitrarily shaped, rigid object at a rate of roughly 10 Hz [14].

4 CONCLUSION

So far, a real-time, low-cost stereo vision system with performance exceeding that of state-of-the-art active scanning laser range finders has remained as one of the unfulfilled goals of computer vision. Commercially available scanning laser range finders, in turn, have been expensive. Fast and cost-effective range sensing at the video rate or higher, such as those presented here, can provide new opportunities in robotics.

Acknowledgements

I express my thanks to my coworkers for the development of the two fast range sensing systems. I worked with Masa Okutomi for developing the theory of multi-baseline stereo. The theory was extensively tested by Tomo Nakahara. The team for video-rate stereo machine development currently consists of Shigeru Kimura, Eiji Kawamura, and Hiroshi Kano. The VLSI-based range sensor development is a joint work with Andy Gruss, Rick Carley, and Shige Tada.

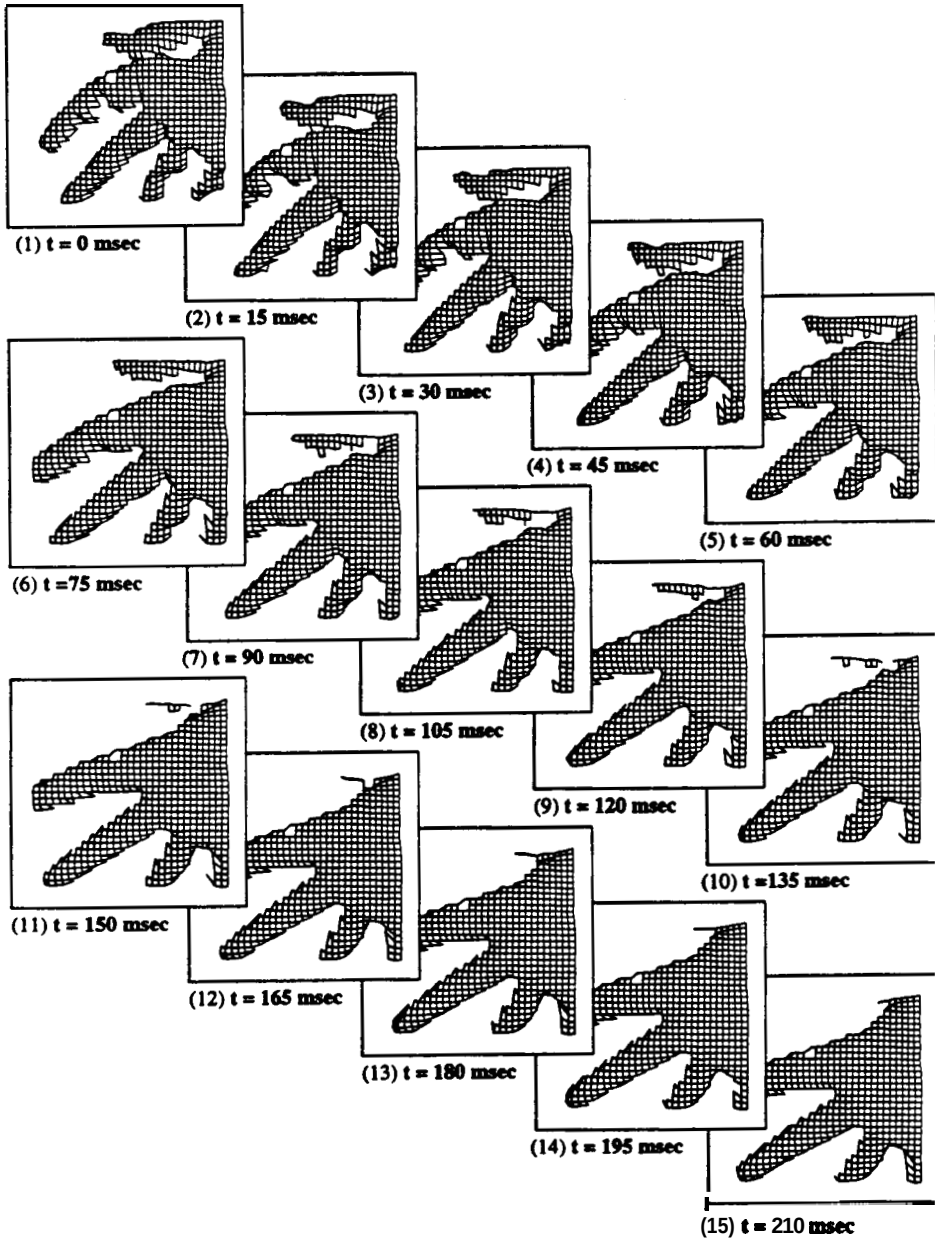

Figure 19: Range-image sequence of a moving hand.

References

[1] K. Araki, Y. Sato, and S. Parthasarathy. High speed rangefinder. In , volume 850, pages 184–188. SPIE, 1987.

[2] N. Ayache and F. Lustman. Trinocular stereovision for robotics. Technical Report 1086, INRIA, Sept 1989.

[3] J. Angelo Beraldin, F. Blais, M. Rioux, J. Domey, and L. Coumoyer. A video rate laser range camera for electronic boards inspection. In , pages 4.1–4.11. Detroit, MI, November 1990.

[4] P. J. Besl. Range imaging sensors. Research Publication GMR-6090, General Motors Research Laboratories, March 1988.

[5] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. Technical Report 1369, Unite de Recherche, INRIA Sophia-Antipolis, Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France, January 1991.

[6] A. Gruss. . PhD thesis, Camegie Mellon University, 13 Nov 1991.

[7] A. Gruss, S. Tada, and T. Kanade. A VLSI smart sensor for fast range imaging. In , Raleigh, NC, July 1992.

[8] T. Kanade, M. Okutomi, and N. Nakahara. A multiple-baseline stereo method. In , pages 409–426. DARPA, January 1992.

[9] A. E. Kayaalp and J. L. Eckman. A pipeline architecture for near real-time stereo range detection. Technical Report GDLS-AI-TR-88-1, General Dynamics AI Lab, November 1988.

[10] L.H. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. , 8 (1):71–91, 1992.

[11] T. Nakahara and T. Kanade. Experiments in multiple-baseline stereo. Technical report, Carnegie Mellon University, Computer Science Department, August 1992.

[12] H.K. Nishihara. Real-time implementation of a sign-correlation algorithm for image-matching. (Draft) Teleos Research, February 1990.

[13] M. Okutomi and T. Kanade. A multiple-baseline stereo. In , June 1991. Also appeared in IEEE Trans. on PAMI, 15(4), 1993.

[14] D. A. Simon, M. Hebert, and T. Kanade. Real-time 3-D pose estimation using a high-speed range sensor. In , 1994.

[15] J. Webb. Implementation and performance of fast parallel multi-baseline stereo vision. In , pages 1005–1012. DARPA, April 1993.

[16] K. Yoshida and S. Hirose. Real-time stereo vision with multiplearrayed camera. Tokyo Instituteof Technology-Precision and Intelligence Laboratory, Department of Mechanical Engineering Science, 199x.